

# Инструменты компании Интел для разработки программного обеспечения.

**Дмитрий Тараканов**

**Инженер по программному обеспечению**

**ISDEF 2004, 17 сентября 2004**

# Intel® Software Development Products

- **Производительность**
  - Главное назначение всех инструментов от Интел
  - Разработчики инструментов работают в непосредственном контакте с разработчиками архитектур процессоров
- **Совместимость**
  - Инструменты Интел совместимы с ведущими IDE
  - Легко интегрируются в процесс разработки

[www.intel.com/software/products](http://www.intel.com/software/products)



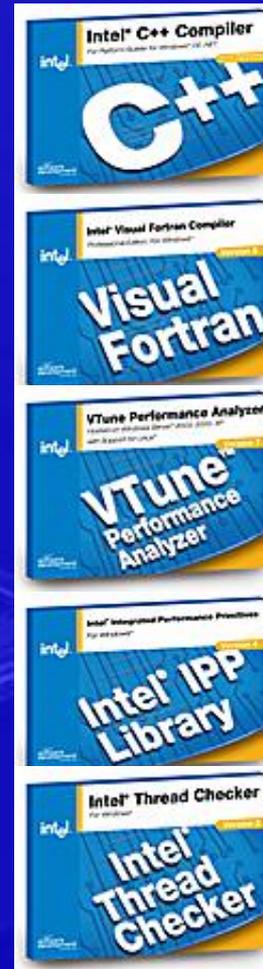
# Intel® Software Development Products

- **Доступность**
  - Windows\* и Linux\*
  - Все Интеловские платформы (оптимизация под каждую платформу)
  - Ранняя доступность (до выпуска процессора)
- **Поддержка**
  - Premier Customer Support мирового уровня
- **Обучение**
  - Intel® Software College
    - Все инструменты и платформы
  - Множество WEB-ресурсов



# Intel® Software Development Products

- **Intel® Compilers**  
Лучший способ добиться высокой производительности приложений на архитектурах Интел
- **Intel® VTune™ Performance Analyzers**  
Быстрый анализ узких мест производительности и способы их разрешения
- **Intel® Performance Libraries**  
Высокооптимизированные библиотеки функций широкого спектра применения – производительность и портируемость
- **Intel® Threading Tools**  
Ускоряют и облегчают разработку многопоточных приложений
- **Intel® Software College**  
Эксперты поделятся знаниями Интеловских архитектур и инструментов



**Intel Software Development Tools for  
Intel® Personal Internet Client Architecture,  
Pentium® M, Pentium® 4, Intel® Xeon™ and  
Itanium® 2 Processors**



# Intel Software Development Products

Intel Software Development Products		 Windows* Linux*		   Windows* Linux*		Intel Personal Internet Client Architecture for Mobile Devices Windows* Linux*	
		Windows*	Linux*	Windows*	Linux*	Windows*	Linux*
<i>Компиляторы</i>	<b>C++</b>	●	●	●	●	●	○
	<b>Fortran</b>	●	●	●	●	●	NA
<i>Анализаторы производ-сти</i>	<b>VTune™ Performance Analyzer</b>	●	●	●	●	●	○
	<b>Math Kernel Library</b>	●	●	●	●	NA	NA
<i>Библиотеки</i>	<b>Integrated Performance Primitives</b>	●	●	●	●	●	●
	<b>Thread Checker/Profiler</b>	○	○	●	○	NA	NA



Поддержка в 2004-2005

NA Not available



***From Supercomputers to Cell Phones,  
Intel Software Development Products Enable  
Application Development Across Intel Processors***

\* Другие наименования и товарные знаки являются собственностью своих законных владельцев.



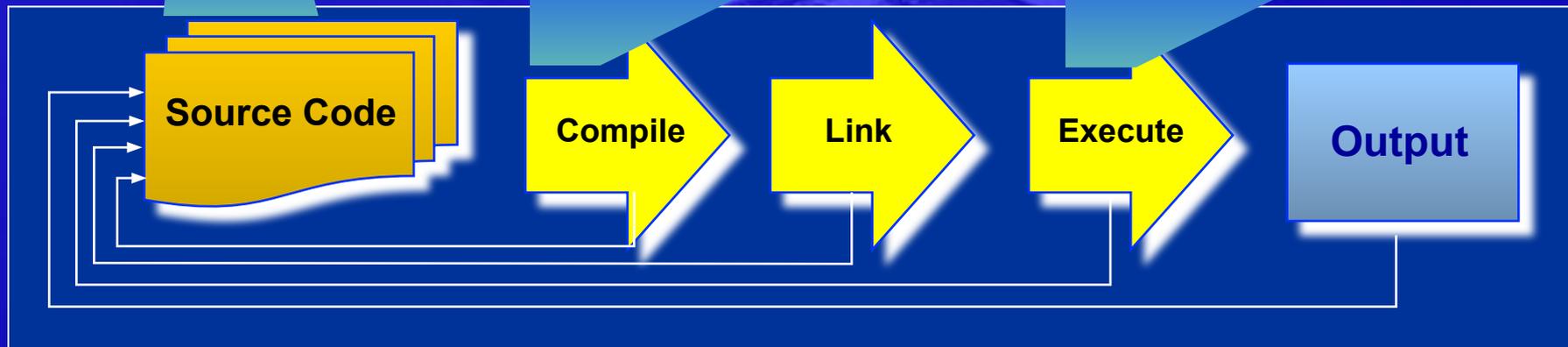
# Intel® Software Development Products поддерживают весь цикл разработки

## Intel® Solution Services

Intel®  
Performance  
Libraries:  
MKL, IPP, GPP

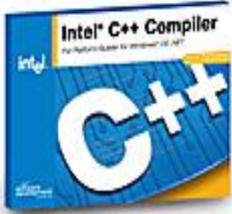
- Intel® C/C++ Compiler
- Intel® Fortran Compiler

- VTune™ Performance Analyzer Tools, Threading Tools



# Семейство компиляторов Интел

Лучший способ добиться высокой производительности приложений на архитектурах Интел

Language	Architecture	OS/Platform	Name / Name since 8.0 release
	IA32	Windows*	icl
		Linux*	icc → ecc
	Itanium®	Windows*	ecl
		Linux*	ecc
	Xscale™	Microsoft eMbedded Visual C++	ccxsce
		Platform Builder for Win CE .NET*	ccxsce
C	IA32 and Itanium®	Windows / EFI Byte Code (EBC) Virtual Machine	iec
	IA32	Windows*	ifl → ifort
		Linux*	lfc → ifort
	Itanium®	Windows*	Efl → ifort
		Linux*	efc → ifort



\* Другие наименования и товарные знаки являются собственностью своих законных владельцев.



# Факторы повышения производительности

- Автовекторизация для архитектуры NetBurst™
- Программная конвейеризация под архитектуру Itanium®
- Параллелизация, основанная на OpenMP\*, для многопроцессорных систем и систем с Hyper-Threading
- Автопараллелизация
- Предварительная выборка данных
- Межпроцедурная оптимизация (IPO)
- Оптимизация по профилированию (PGO)
- Диспетчеризация ЦП во время исполнения

**Генерация процессор-специфичного кода  
Единый исходный код для  
всего семейства процессоров Интел**

# Встроенные средства SIMD-расширений

- встроенные средства SIMD-расширений работают с упакованными данными до 128 байт в длину, что обеспечивает возможность параллельной обработки элементов данных
- позволяют использовать Си функции вместо кодирования на языке ассемблера
- обеспечивают доступ к основным возможностям, нереализуемым с применением обычных методик кодирования
- большинство команд в рамках технологии MMX™, SSE и SSE2 имеют соответствующие встроенные функции на языке Си

# Автовекторизация

- Автоматически применяет SIMD команды в наборах команд SSE2, SSE и MMX™
- Определяет операции программы, которые можно выполнять параллельно, после чего конвертирует последовательную программу для обработки 2, 4, 8 или 16 элементов за одну операцию в зависимости от типа данных
- Все стандартные математические функции в Си имеют SIMD реализации
- Достаточно указать ключ в командной строке

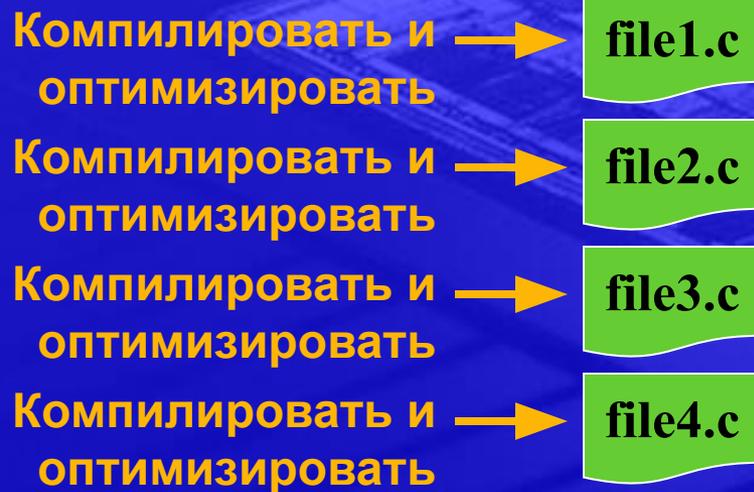
# Программная конвейеризация

- Программная конвейеризация предназначена для перекрытия итераций циклов
- Использует мощную поддержку программной конвейеризации, обеспечиваемую архитектурой Itanium®
  - циклический сдвиг регистров
  - специальные команды ветвления для циклов
  - большой массив регистров
- Компилятор работает автоматически без необходимости указания каких-либо ключей в командной строке

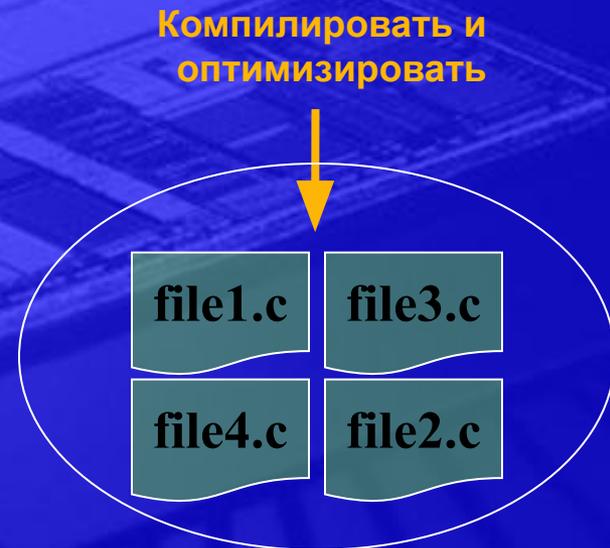
# Межпроцедурная оптимизация

Распространяет оптимизацию на все файлы

## Без IPO



## С IPO



# Оптимизация по профилированию

- Оптимальна для кода с часто выполняемыми ветвлениями, которые трудно предсказать во время компиляции
- Оптимизирует принятие компилятором решений о подстановке функций
- Включает следующие стадии
  - Инструментальная компиляция и связывание
  - Запуск полученного файла для создания файлов динамической информации (.dyn)
  - Компиляция с использованием .dyn файла.

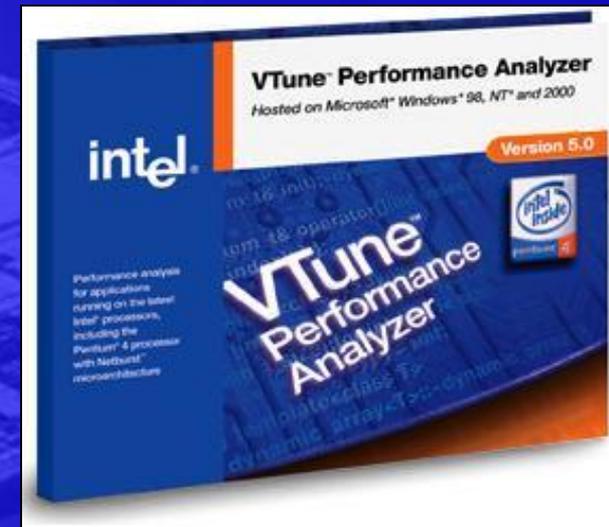
# Поддержка многопоточной разработки в компиляторах Intel®

- **Поддержка OpenMP\* в компиляторах Intel®**
  - Предоставляет стандартный набор библиотечных функций для упрощения управления программой в режиме параллельного исполнения
  - Обеспечивает расширение библиотеки OpenMP\* для работы с памятью в многопоточном режиме
- **Автопараллелизатор компилятора Intel®**
  - Обнаруживает циклы, которые могут безопасно выполняться в параллели, и автоматически генерирует многопоточный код для подобных циклов
  - Освобождает пользователя от необходимости заниматься низкоуровневыми задачами по декомпозиции итераций, совместному использованию данных, планированию и синхронизации потоков
  - Обеспечивает повышение производительности для многопроцессорных систем
- **Работает не только на многопроцессорных машинах, но и на однопроцессорных Pentium® 4 с технологией Hyper-Threading**

# VTune™ Performance Analyzer

Быстрый анализ узких мест производительности и способы их разрешения

- Помогает определить и локализовать проблемы производительности ПО посредством:
  - Сбора широкого спектра показателей производительности с ОС на которой выполняется Ваше приложение
  - Обработки и отображения данных в различных видах, начиная с system-wide и заканчивая исходным кодом и процессорными инструкциями
  - Идентификации потенциальных проблем производительности и предложения вариантов их разрешения



Поддерживает все современные процессоры Intel®, включая Itanium® 2, Pentium® 4, Pentium® M и XScale™



# VTune™ Performance Analyzer

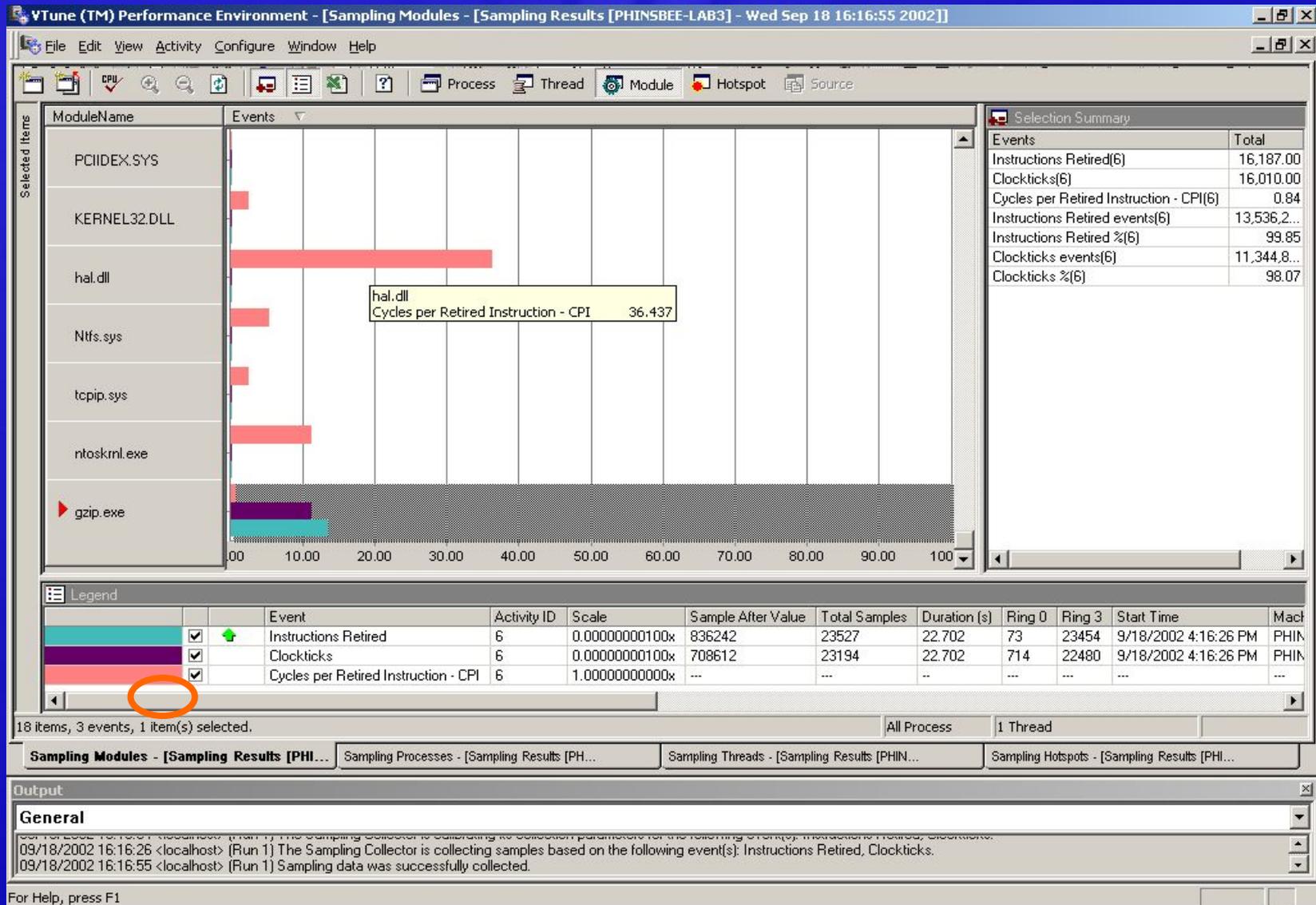
- Поддерживает
  - Платформы:
    - Семейство Intel® IA-32
    - Семейство Itanium®
    - XScale®
  - Операционные системы:
    - Microsoft Windows\*
    - Linux\* (Red Hat, SuSe и др.)
  - MRTE
    - Java (BEA, IBM, Microsoft, Sun)
    - .NET
- Поддерживает локальный и удалённый сбор данных
  - Агент и коллекторы данных на удалённой машине
    - Microsoft Windows\*
    - Linux\*
    - MTRE
  - Управление процессом, анализ и отображение результатов на host-машине
- Поддерживает интерфейс командной строки
- Интегрируется в Visual Studio 7

# Обзор функциональности

- **Sampling**
  - time-based
  - event-based
  - Overtime view
- **Call graph**
  - Граф вызовов с подробной информацией о временных затратах
  - Критический путь исполнения
- **Counter monitor**
  - Системные счётчики производительности
- **Статический анализ исполняемых модулей**
- **Intel® Tuning Assistant**
  - Комментарии по проблемам, подсказки по модификации кода
- **Обработка данных**
  - «Мастера» для конфигурирования коллекторов
  - Сравнение и слияние результатов
  - Упаковка и перенос проектов на другую машину
- **Getting Started tutorial**

# Sampling Process View

Данные по всей системе



VTune and the Intel logo are trademarks or registered trademarks of Intel Corporation or its subsidiaries in the United States or other countries.



# Sampling Source View

Показатели производительности по линиям исходного кода

The screenshot displays the VTune Performance Environment interface. The main window shows the 'Source View' for a file named 'deflate.c'. The source code is displayed with columns for 'Address', 'Line', 'Source', 'Instructions Retired', and 'Clockticks (6)'. The code includes a loop that processes match lengths, with a 'while' loop that continues as long as the match length is greater than a limit and the chain length is not zero. The 'Instructions Retired' and 'Clockticks (6)' columns show the performance metrics for each line of code.

Address	Line	Source	Instructions Retired	Clockticks (6)
459				
0x152D	460	len = MAX_MATCH - (int)(strend - scan);	17	17
0x152F	461	scan = strend - MAX_MATCH;	75	104
462				
463		#endif /* UNALIGNED_OK */		
464				
0x153D	465	if (len > best_len) {	9	18
466		match_start = cur_match;		
467		best_len = len;		
0x1541	468	if (len >= nice_match) break;	26	30
469		#ifdef UNALIGNED_OK		
470		scan_end = *(ush*)(scan+best_len-1);		
471		#else		
0x1552	472	scan_end1 = scan[best_len-1];	3	3
0x1556	473	scan_end = scan[best_len];	18	15
474		#endif		
475		}		
476		} while ((cur_match = prev[cur_match & WMASK]) > limit		
0x155D	477	&& --chain_length != 0);	7,776	7,118
478				
0x1586	479	return best_len;	188	207
0x158C	480	}	12	9

Below the source view, there is a 'Function Summary' table and a 'Sampling Results' table. The 'Function Summary' table shows the address, size, and function name for each function. The 'Sampling Results' table shows the instructions retired, clockticks, and cycles per retired instruction for each function.

Address	Size	Function	Class	Instructions Retired (6)	Clockticks (6)	Cycles per Retired Instruction - ...
-----	-----	--- Selected Range ---	-----	18	25	1.177
0x1320	0x119	lm_init		0	0	
0x1440	0x150	longest_match		12,891	12,215	0.803
0x1590	0xFD	fill_window		574	684	1.010
0x1690	0x305	deflate		1,045	1,312	1.064
0x1010	0x238	deflate_fast		0	0	

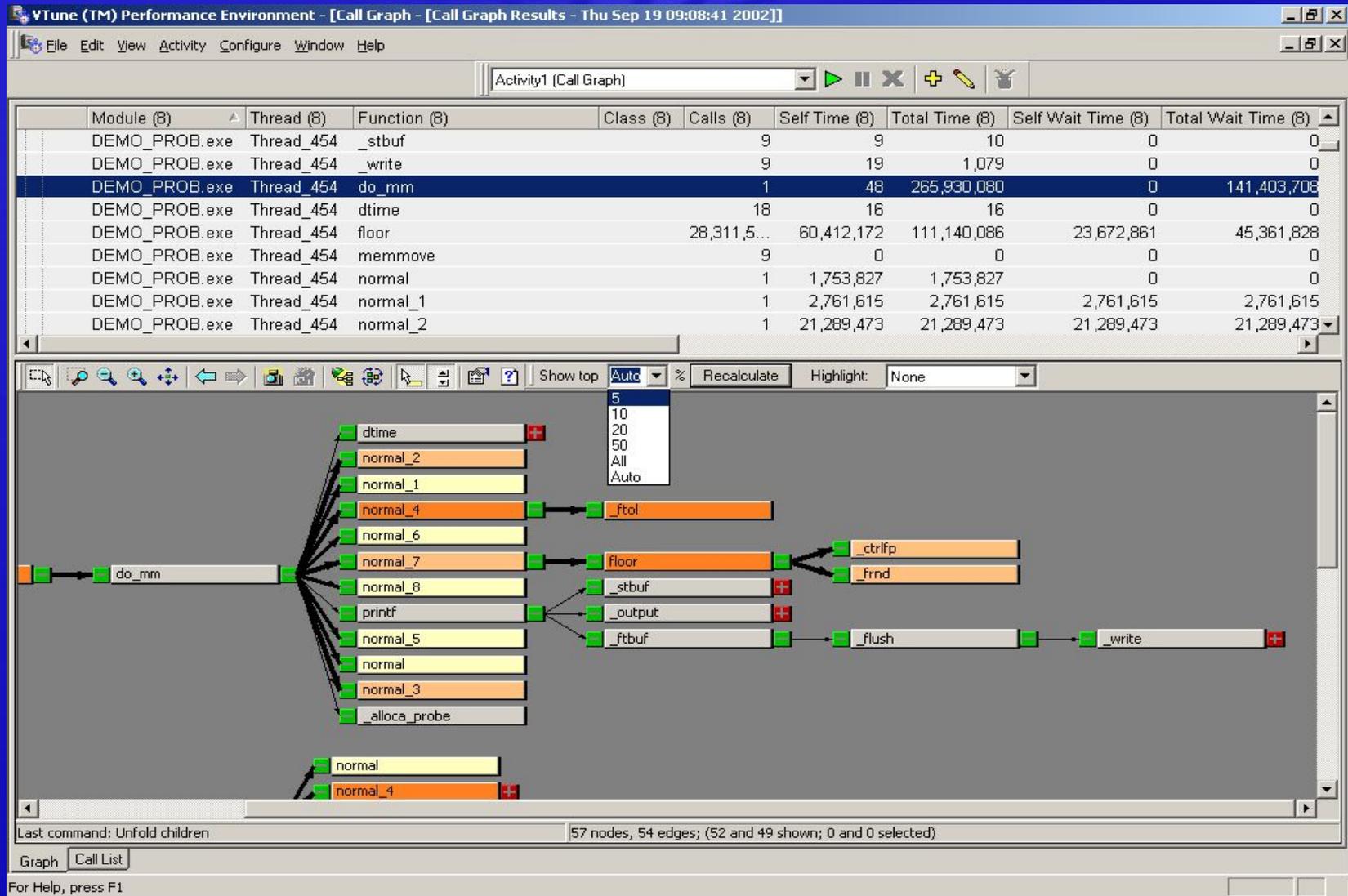
The 'Output' window shows the following text:

```
General
09/18/2002 16:16:26 <localhost> (Run 1) The Sampling Collector is collecting samples based on the following event(s): Instructions Retired, Clockticks.
09/18/2002 16:16:55 <localhost> (Run 1) Sampling data was successfully collected.
```



# VTune™ Call-Graph View

Граф вызовов функций, критический путь исполнения



# VTune™ для Linux\* – две возможности Локальный и удалённый сбор данных

- **VTune™ CLI2.0** – устанавливается локально на Linux машине
  - Интерфейс командной строки
  - Следующая версия будет графической
    - Будет доступна в этом году
- **VTune™ 7.1 Windows**
  - Коллектора на Linux машине для удалённого сбора данных
  - Просмотр данных на Windows host-машине
  - GUI

# VTune™ Performance Analyzer 2.0 for Linux\*

- Поддерживает
  - Платформы:
    - Семейство Intel® IA-32
    - Семейство Itanium®
  - Операционные системы:
    - Linux\* (Red Hat, SuSe и др.)
  - MRTE
    - Java (BEA, IBM, Sun) на IA-32
- Интерфейс командной строки
- Результаты совместимы с Windows-версией.
  - Могут быть упакованы и просмотрены с помощью Windows-версии анализатора
- Поддерживает до 64 процессоров
- Обеспечивается высоким уровнем клиентского сервиса



\* Другие наименования и товарные знаки являются собственностью своих законных владельцев.



# Обзор функциональности

- **Sampling**
  - event-based
- **Call graph**
- **Source view**
- **Обработка данных**
  - Упаковка и перенос проектов на другую машину
- **Map страницы и HTML руководство**

# VTune™ CLI2.0 - sampling view

```
vtune@nntvtune117-SuSe82:/tmp/tests/temp> vtl view
VTune(TM) Performance Analyzer 2.0 for Linux*
Copyright (C) 2000-2003 Intel Corporation. All rights reserved.
```

## Event Summary

### Instructions Retired

```
12465 = Samples collected due to this event
839270 = Sample after value used during collection
10461500550 = Total events (samples*SAV)
```

### Clockticks

```
12475 = Samples collected due to this event
985415 = Sample after value used during collection
12293052125 = Total events (samples*SAV)
```

## Module View (all values in decimal)

Module Event	Process	Events%	Samples	Events	Module Path
vtunedemo	vtunedemo				
Instructions Retired		99.60%	12415	10419537050	/tmp/tests/
Clockticks		97.80%	12201	12023048415	
libc.so.6	vtunedemo				
Instructions Retired		0.30%	37	31052990	/lib/libc.so
Clockticks		0.66%	82	80804030	
vmlinux	Pid 0x0				
Instructions Retired		0.02%	2	1678540	vmlinux
Clockticks		0.83%	104	102483160	
+0ther32+	XFree86				
Instructions Retired		0.01%	1	839270	
Clockticks		0.13%	16	15766640	
vmlinux	vtl.bin				
Instructions Retired		0.01%	1	839270	vmlinux
Clockticks		0.07%	9	8868735	

# VTune™ CLI2.0 - source view

```
23      0  0  float a[100];          // scratchpad areas for calculations
24      0  0  float b[100];
25      0  0
26      0  0  int Global_Test_if_putp; // global value used in test_if() functions
27      0  0
28      0  0  int Flag_SelfExit;     // if -exit on cmdline, exits automatically
29      0  0
30      0  0
31      0  0
32      0  0  // Walks matrix performing simple function operations
0x3a4 33      0  7  void test_if(int *a,int *p, int *q, int n) {
34      0  0      int i;
35      0  0
0x3ad 36      1024 752      for(i = 0; i < n; i++)
0x3be 37      560  370      if(Global_Test_if_putp == 1)
0x3c7 38      1435 911      a[i] = p[i] + q[i];
39      0  0      else
0x3f9 40      1399 806      a[i] = p[i] - q[i];
41      0  0
0x432 42      121  83  }
```

# Intel® Threading Tool Kit

- **Инструменты отладки и мониторинга многопоточных приложений**
  - Thread Checker: проверяет корректность многопоточных приложений
  - Thread Profiler: анализирует производительность в аспекте организации потоков
- **Устанавливаются на VTune ( 7.0 или 7.1 )**
- **Поддерживают**
  - **Платформы:**
    - Семейство Intel® IA-32
    - Для Itanium в процессе разработки
  - **Операционные системы:**
    - Microsoft Windows\*
    - Для Linux в процессе разработки
- **Поддерживают потоковые модели:**
  - OpenMP\*
  - Windows\* Threading API

# Intel® Thread Checker 2.0

## Различные режимы анализа

- **Инструментирование исполняемых файлов**
  - Сбор данных осуществляется внутри оболочки VTune™
  - Требует наличия только исполняемых файлов программы; VTune™ управляет всем процессом
  - Позволяет анализировать (системные) библиотеки даже при отсутствии исходного кода
  - Не требует специальной (пере)компиляции или (пере) сборки
- **Инструментирование на этапе компиляции**
  - Компилировать с icl /Qtcheck
  - Более качественная информация по местонахождению в исходном коде переменных, выражений и т.д.
  - Производительность (сопоставление компиляции и инструментирования, анализ)

# Intel® Thread Checker

## Список ошибок

- Настраиваемый
- Связь с исходным кодом

## Представление исходного кода

- Контекст ошибки
- Место ошибки
- Отслеживание стека вызовов функции

The screenshot displays the Intel Thread Checker Performance Environment interface. The top window, titled "Intel® Thread Checker Results - Wed Jul 24 11:06:11 2002: Error List", shows a table of detected errors. The table has columns for Description, 1st Access [Best], 1st Access [Variable], 2nd Access [Best], and 2nd Access [Variable]. The errors are grouped into three categories: Group 1 (R -> W data-race), Group 2 (Thread Info), and Group 3 (W -> W data-race). The bottom window, titled "Intel® Thread Checker Results - Wed Jul 24 11:06:11 2002 (ID=0)1st Access", shows a stack trace for the first access of the error. The stack trace is for the function RadialBlur in the file "RadialBlur.cpp" at line 103. The code snippet shows a nested loop for pixel iteration, with the error occurring at line 103 where the x-coordinate is calculated.

Description	1st Access [Best]	1st Access [Variable]	2nd Access [Best]	2nd Access [Variable]
<b>Group 1: R -&gt; W data-race (anti-dependence)</b>				
R -> W data-race (anti-dependence)	"RadialBlur.cpp":115	x	"RadialBlur.cpp":103	x
R -> W data-race (anti-dependence)	"RadialBlur.cpp":116	y	"RadialBlur.cpp":104	y
R -> W data-race (anti-dependence)	"RadialBlur.cpp":128	xr	"RadialBlur.cpp":115	xr
R -> W data-race (anti-dependence)	"RadialBlur.cpp":128	yr	"RadialBlur.cpp":116	yr
R -> W data-race (anti-dependence)	"RadialBlur.cpp":120	r	"RadialBlur.cpp":117	r
R -> W data-race (anti-dependence)	"RadialBlur.cpp":125	step	"RadialBlur.cpp":120	step
R -> W data-race (anti-dependence)	"RadialBlur.cpp":138	arcPixelOffset	"RadialBlur.cpp":136	arcPixelOffset
R -> W data-race (anti-dependence)	"RadialBlur.cpp":106	x	"RadialBlur.cpp":103	x
R -> W data-race (anti-dependence)	"RadialBlur.cpp":106	y	"RadialBlur.cpp":104	y
<b>Group 2: Thread Info - includes stack usage</b>				
<b>Group 3: W -&gt; W data-race (output dependence)</b>				
W -> W data-race (output dependence)	"RadialBlur.cpp":103	x	"RadialBlur.cpp":103	x
W -> W data-race (output dependence)	"RadialBlur.cpp":104	y	"RadialBlur.cpp":104	y
W -> W data-race (output dependence)	"RadialBlur.cpp":115	xr	"RadialBlur.cpp":115	xr
W -> W data-race (output dependence)	"RadialBlur.cpp":116	yr	"RadialBlur.cpp":116	yr

```
Stack Trace: RadialBlur(YAXPAXJ0JPAURect "RadialBlur.cpp":103)
Line 98 for (pixelY = 0; pixelY < rectHeight; pixelY++)
Line 99 {
Line 100     for (pixelX = 0; pixelX < rectWidth; pixelX++)
Line 101     {
Line 102         // Find (x,y) coordinates of current pixel
Line 103         x = gFilterRecord->inRect.left + pixelX;
Line 104         y = gFilterRecord->inRect.top + pixelY;
Line 105
Line 106         pixelOffset = (x * gFilterRecord->outColumnBytes) +
Line 107                     (y * gFilterRecord->outRowBytes);
Line 108
Line 109         bool leaveItAlone = false;
Line 110         if (maskPixel != NULL && !(*(maskPixel + pixelOffset)) && !gParams-
Line 111             leaveItAlone = true;
```

# Intel® Thread Profiler 2.0

Для OpenMP\*

- Собирает и отображает данные по производительности в аспекте организации потоков
  - Упрощает процесс отладки распараллеливания
  - Представляет данные по каждому потоку и региону распараллеливания
  - Затраты на синхронизацию, накладные расходы системы
  - Дисбаланс по нагрузке
  - Сравнение результатов разных запусков
  - Аппроксимация масштабируемости приложения

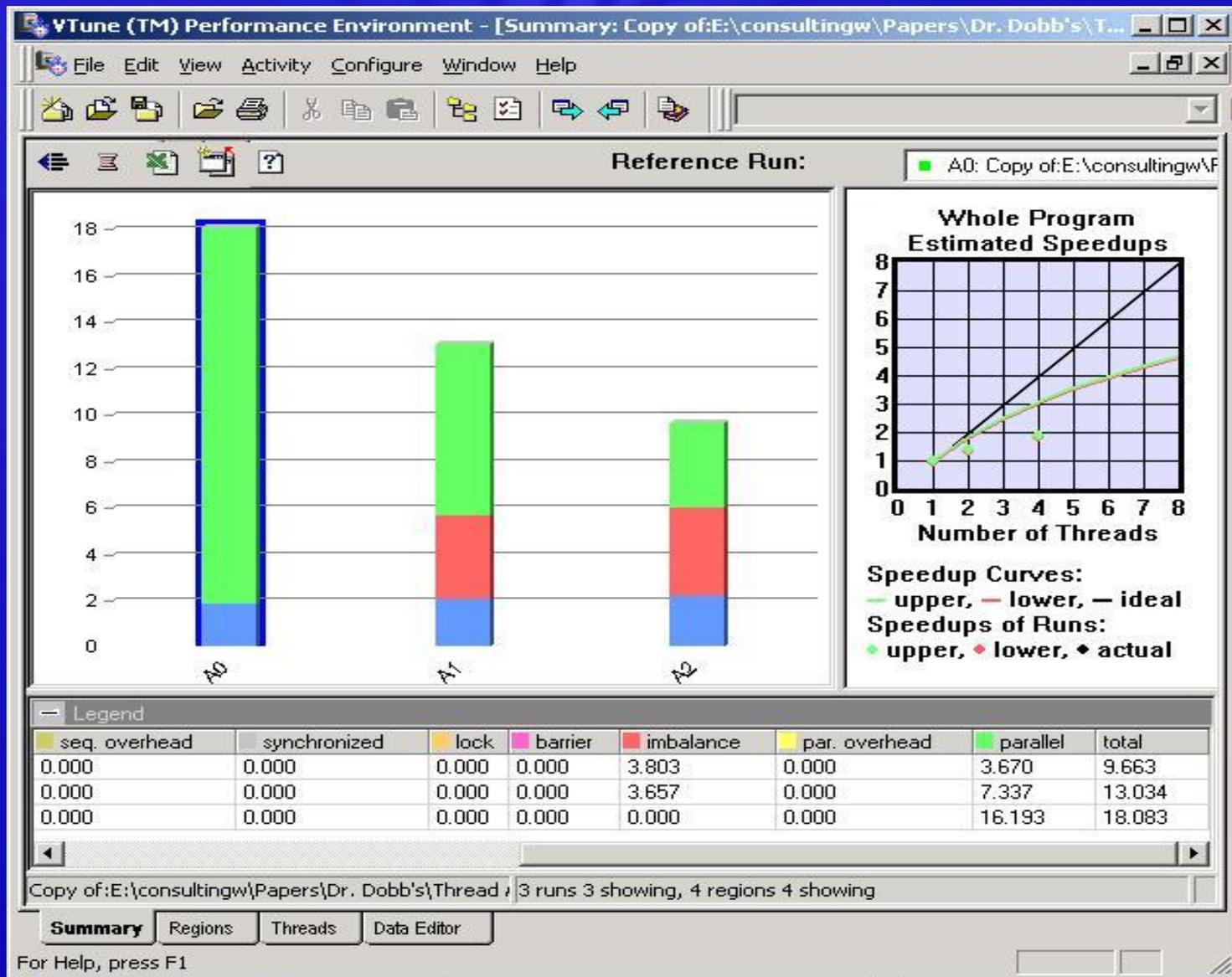
# Intel® Thread Profiler 2.0

Для потоков Windows\*

- **Отображает данные:**
  - Критические пути исполнения программы
  - Распределение временных затрат по критическому пути на:
    - Исполнение
    - Синхронизацию
    - Ожидание
    - Блокирование
  - Аналогичные данные по потокам программы и уровню параллелизма
  - Ассоциации потоковых событий со строками исходного кода

# Intel® Thread Profiler 2.0

## Для OpenMP\*

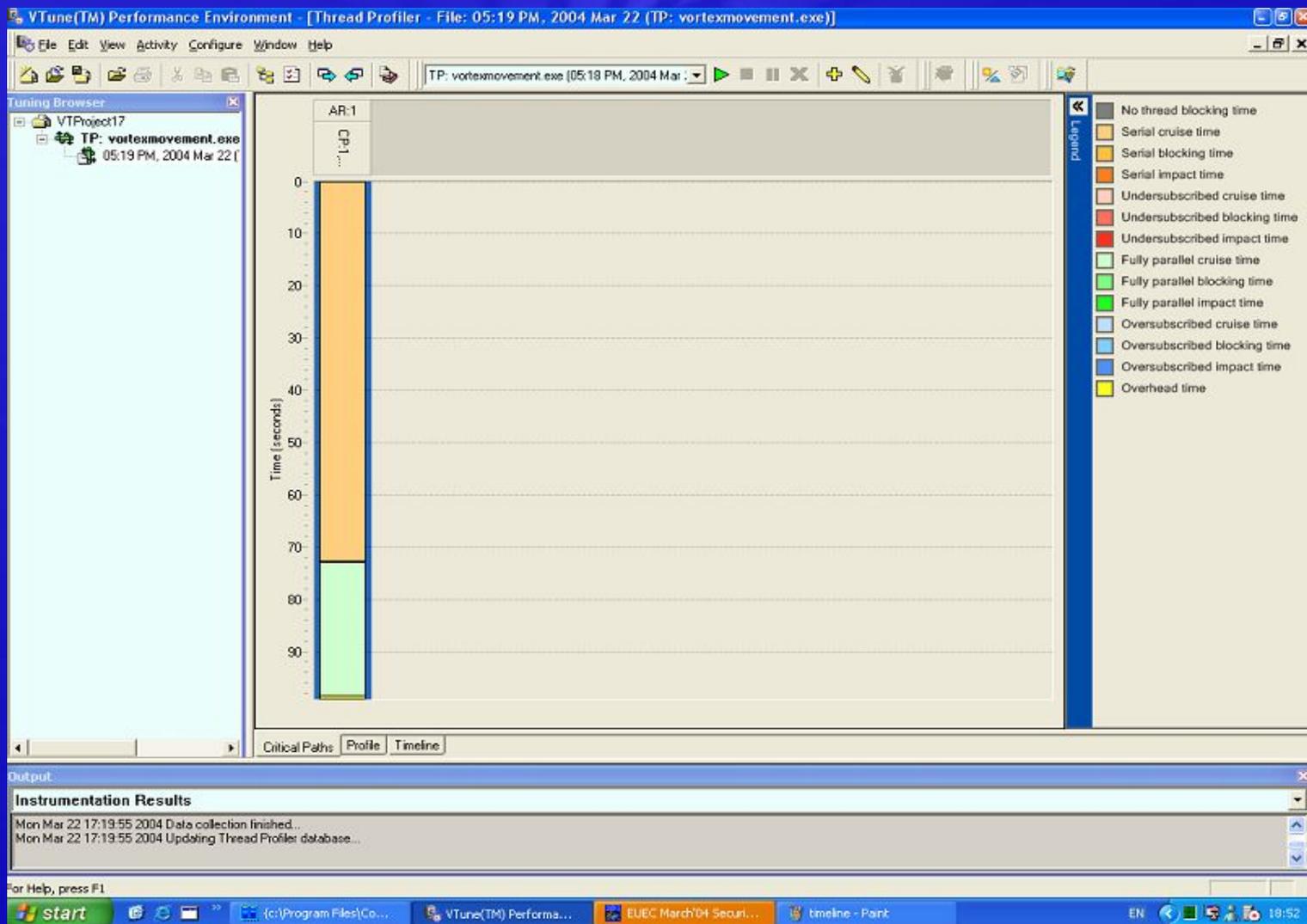


\* Другие наименования и товарные знаки являются собственностью своих законных владельцев.



# Intel® Thread Profiler 2.0

## Для потоков Windows\*

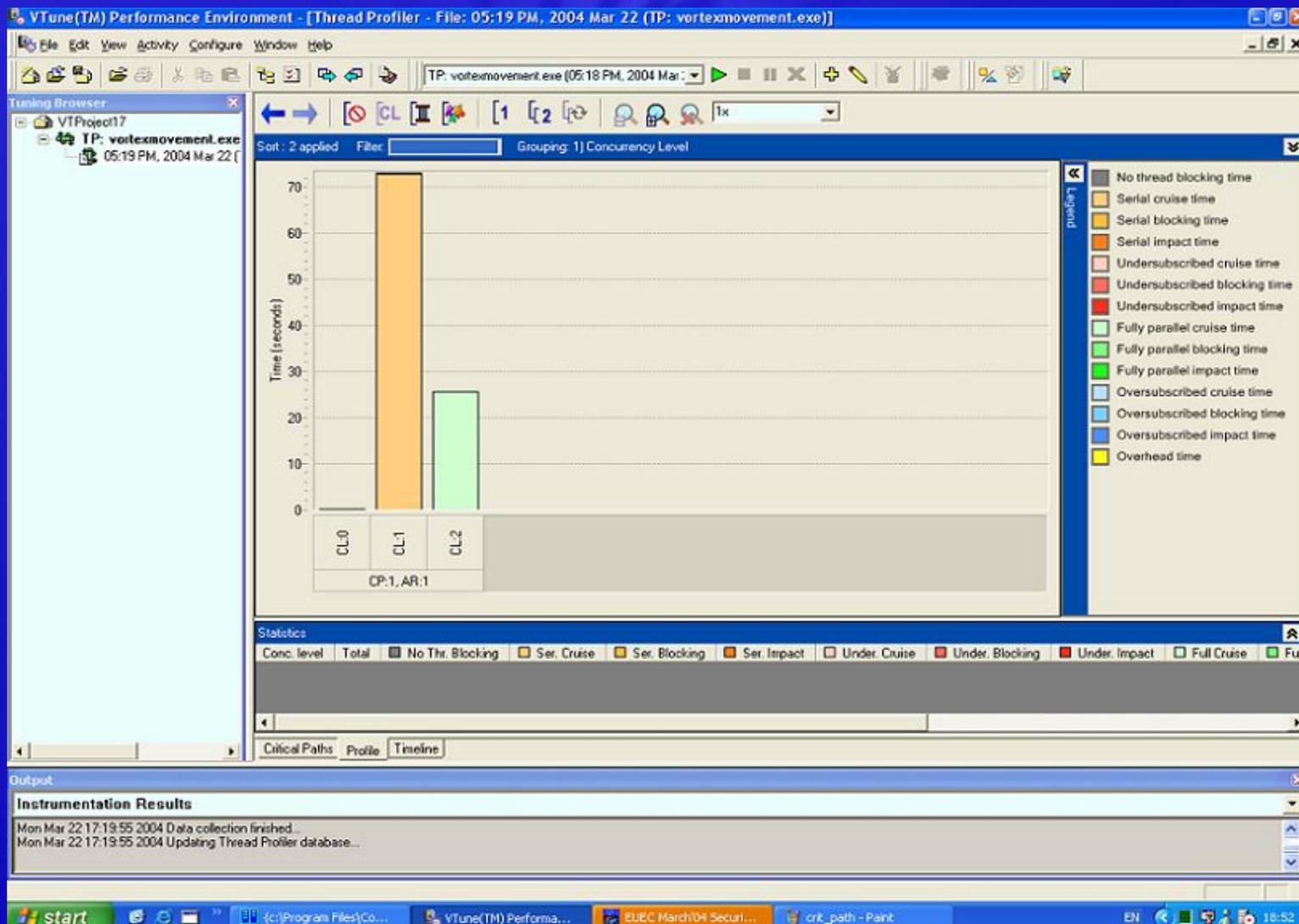


\* Другие наименования и товарные знаки являются собственностью своих законных владельцев.



# Intel® Thread Profiler 2.0

## Для потоков Windows\*



\* Другие наименования и товарные знаки являются собственностью своих законных владельцев.



# Intel® Performance Libraries

Высокооптимизированные библиотеки функций широкого спектра применения – производительность и портируемость

## MKL – Intel® Math Kernel Library

- Linear Algebra: LAPACK plus BLAS (Levels 1, 2, 3)
- Discrete Fourier Transforms (DFT)
- Vector Statistical Library functions (VSL)
- Vector transcendental math functions (VML)

## IPP – Intel® Integrated Performance Primitives

- signal, image, graphic, multimedia and numeric processing functions

- Доступны для Linux\* and Windows\*
- Индивидуально оптимизированы под
  - Pentium® III, Pentium® 4
  - Itanium®
  - IPP также под StrongARM\*, Xscale™
- Во время исполнения загружается код, оптимальный для данного процессора
- Избавляют разработчиков от написания процессор-специфичного кода

## GPP - Intel® Graphics Performance Primitives

- Доступны для PDA/Xscale™ с PPC 2002
  - Поддержка других ОС в процессе разработки



# Ресурсы

На <http://developer.intel.com>

- On-line курсы и документация
- Руководства по архитектуре процессоров
- Открытые коды
- Новые релизы / патчи
- Объявления, новости

**Intel Press предлагает множество полезных книг**

# Intel® Developer Services

	Ресурсы	Продукция и услуги	Решения	Технологии и тенденции	Поддержка
--	---------	--------------------	---------	------------------------	-----------

## Разработчикам программного обеспечения

Получите предварительный доступ к платформам Intel®

Продукция Intel® для разработки ПО

Микропроцессоры

Сети и коммуникации

Каталог решений

Загружаемые файлы

Отрасли промышленности

Операционные системы

Технологии ПО

Стратегии и решения

Колонки

Новости и события

Сообщество

Обучение

Войти:

Intel® Developer Services

- Войти
- Регистрация

Поиск ПО:

Поиск

[Подсказки](#) | [Расширенный поиск](#)

Выбрать язык:

Intel® Developer Services



## ПО и решения Intel® Developer Services

### Мобильность: Новая задача, стоящая перед разработчиками ПО с открытым исходным кодом

Узнайте больше о ресурсах, которые помогут Вам воспользоваться преимуществами ПО для мобильных устройств и ПО с открытым исходным кодом.

[ПО для мобильных ПК](#)

- Получите бесплатный компакт-диск с пробными версиями средств разработки на выставке **Softool** 28 сентября – 2 октября, ВВЦ, Москва. Посетите стенд Intel, ВВЦ, павильон №69, на выставке Softool, чтобы ознакомиться с демонстрацией средств разработки Intel® и получить бесплатный компакт-диск с пробными версиями средств разработки!
- Форум независимых разработчиков ПО** 16-19 сентября, гостиница Холидэй Инн Виноградovo. Корпорация Intel выступит в качестве золотого спонсора Форума независимых разработчиков ПО (ISDEF). На этом Форуме Вы сможете пообщаться с независимыми программистами, а также с представителями компьютерных изданий, архивов программного обеспечения, регистрационных органов и другими представителями индустрии программного обеспечения.
- Форум Intel® для разработчиков** 19-20 октября, Российская Академия Наук, Москва, Ленинский проспект, 32a. Получите последнюю информацию по

### Информационный бюллетень

### Early Access Program

Получите доступ к сведениям о новом оборудовании, инструментах, поддержке и маркетинговой информации

[Подробнее >](#)

### Центры разработки

- Процессоры Itanium®
- Процессоры Intel® Xeon™
- Web-сервисы
- Игры
- Стратегия и решения
- Поточная обработка

[Подробнее >](#)

Присоединяйтесь к сообществу

Intel  
developer  
services

intel®

# Итоги

- Intel® Software Developer Tools: широчайшие возможности для создания высокоэффективных приложений любой сложности
- Главные факторы: производительность, ещё раз производительность, плюс совместимость
- Доступность на Linux\*/Windows\* -- C++/Fortran -- IA32/Itanium/XScale
- Постоянное расширение сферы охвата ( HPC, Linux\* )
- Профессиональный сервис мирового уровня

The image features a central, isometric view of a square microchip. The chip's surface is detailed with various rectangular blocks and intricate circuit patterns, all rendered in a monochromatic blue color. The word "DEMO" is prominently displayed in the center of the chip in a bold, yellow, sans-serif font. The background consists of a dark blue gradient with faint, stylized circuit traces that radiate from the center, creating a sense of depth and technological complexity.

**DEMO**

**Спасибо за внимание!**